

Pool Club Management Software – Final Report



Created by: Evan Lyle

AIT 725 Software Engineering Case Study

Advisor: Prof. Jal Irani

December 7, 2023

Contents

I. Introduction	2
II. Technologies.....	3
III. Diagrams	4
III-a. Entity Relationship Diagram (ERD).....	4
III-b. Wireframe	4
III-c. Use Cases	9
IV. Functionality.....	14
V. Challenges	21
V-a. Programming Limitations.....	21
V-b. Time Constraints	23
VI. What I Wish I Knew at the Beginning of the Project.....	26
VII. Future Plans for the Pool Club Management Software	26
VIII. Conclusion.....	27
IX. Acknowledgements.....	27
References.....	28
Appendix A – Proposal	29
Appendix B – Requirements Document	41
Appendix C – API Documentation	41
Appendix D – User Manual	41
Appendix E – Source Code	41

I. Introduction

There is a need to develop an application to assist in swimming pool club management using modern technologies such as Python and MySQL. Too many pool clubs around the country have been plagued with mismanagement, causing irregular hours for pool staff and the management of chemicals (like the pH levels) to be ignored. Many pool clubs exceed their budget due to overusing pool chemicals, running short of pool chemicals, and mismanaging lifeguard scheduling. Additionally, pools are frequently inspected by local health departments. Many times, pools are not prepared for these inspections due to incomplete or missing documentation concerning water testing and licensing certifications resulting in temporary closure. This project has produced an application that keeps track of an application that helps to keep track of your resources.

The reason I chose this project was for a number of learning opportunities, as well as acting as a culmination of all my skills throughout my time as a student of many Applied Information Technology (AIT) courses. The project also demonstrates how technology is a great asset for keeping track of resources and scheduling as long as it is planned out and constructed properly.

I learned how to integrate Python code with MySQL databases, which was done from the belief that learning about cross-language and cross-platform communication is useful so that certain functions already completed in one language can be reused in the context of another language or platform. This saves time that would have otherwise been wasted on recreating certain features from scratch. It also demonstrates how different types of software can communicate with one another in order to avoid compatibility issues. I also learned how to craft a Graphical User Interface (GUI) interface using Python tkinter, a part of Python which I have not interacted with prior, but I feel would be useful given the demand for Python that I have seen in various job announcements online. In addition, I utilized the Python programming language given said recent demand for it.

As for the scope of the project and the size, I believed that it fit the criteria of a graduate case study project due to the involvement of system integration between different software tools and technologies that allow the application to fully unleash its true potential. It is also a worthy project since pool maintenance has a lot of moving parts and different variables attached to it depending on whether or not you have a public or private pool, much like how different maintenance operations have different needs and data needing to be managed depending on the type of maintenance being tracked.

In this document, you will learn about the journey I took in creating the pool club management software; what I planned to do with it; what I struggled with; what I ran out of time to implement; and future plans for this software.

II. Technologies

During the planning phase of my project, I used dia (all lowercase title) to develop my Entity Relationship Diagram (ERD), Balsamiq to develop the wireframes, and Microsoft Visio to develop my use case diagrams. The technologies I have used in my program include Python, Python tkinter, MySQL, MySQL Connector/Python, and GitHub.

For Python, it is the main programming language I used to create the Application Programming Interfaces (APIs) and the Graphical User Interfaces (GUIs). Python tkinter is a Python interface editor technology that allows Python to build GUIs, which I did for most corresponding API files (with two exceptions for reasons that will be covered later in this report).

MySQL Connector/Python is a connector tool used for connecting the Python programming language to communicate with the MySQL database. Without this tool, the Pool Club Management Software would not be possible.

The GitHub repository is used to store different versions of the same data and code, as well as provide specific information on changes or updates made to the project. Microsoft Word was used for writing the different documents, which was then converted to Adobe PDF files. The water quality regulation reports are also output by the program in the PDF file format. Last but not least, Microsoft Visual Studio Code was the platform used to write and execute the Python code for this project.

For more information, please refer to the user manual.

III. Diagrams

III-a. Entity Relationship Diagram (ERD)

Below is the ERD that I used in the planning phase of the project. The diagram illustrates the organization of the database and how entities are organized and their relationships.

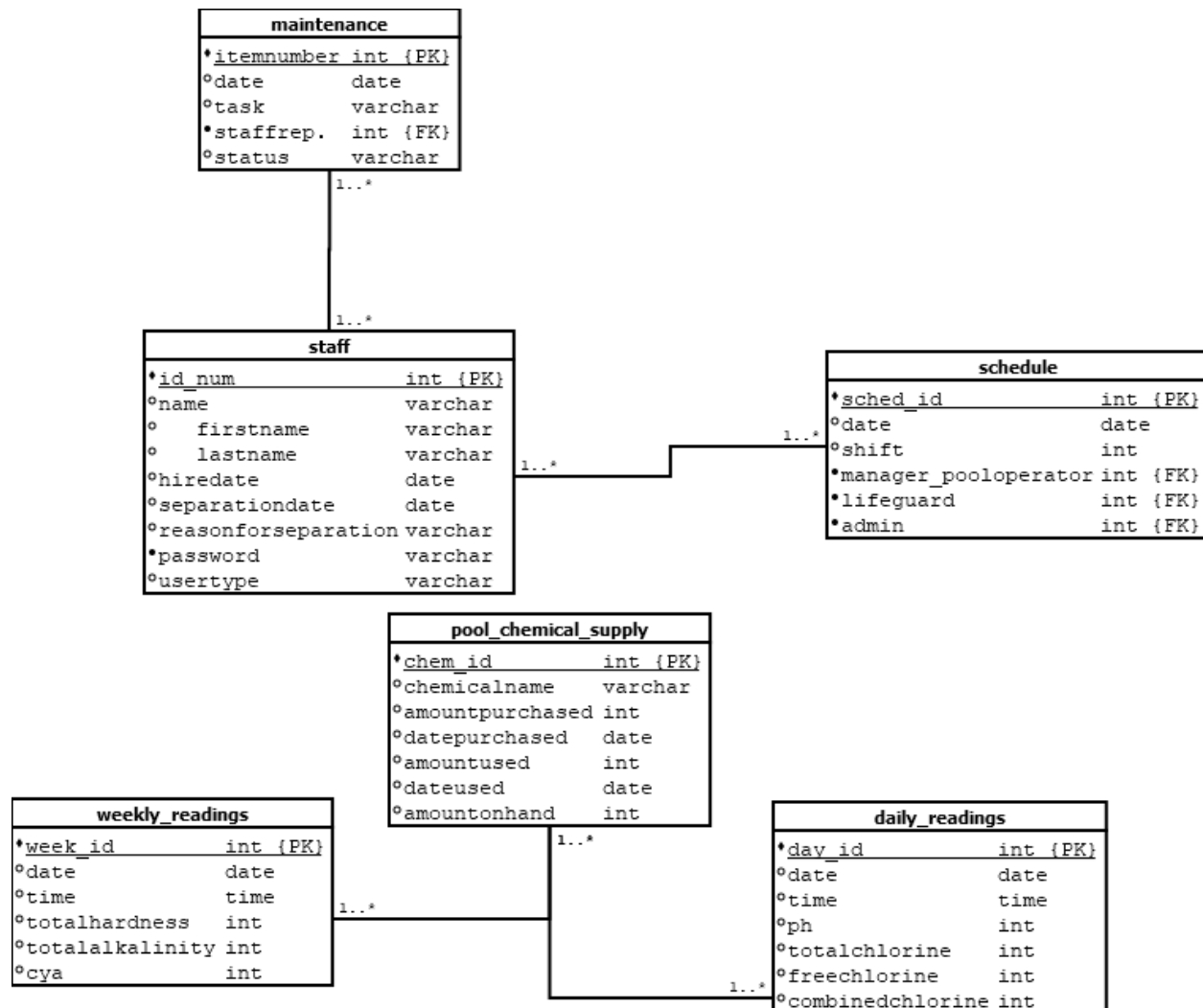


Figure 1 – Entity Relationship Diagram

III-b. Wireframe

The wireframes below are the mockup sketches also developed during the planning phase. They represent the initial vision of each of the screens currently present in the program. I developed them using the Balsamiq software.

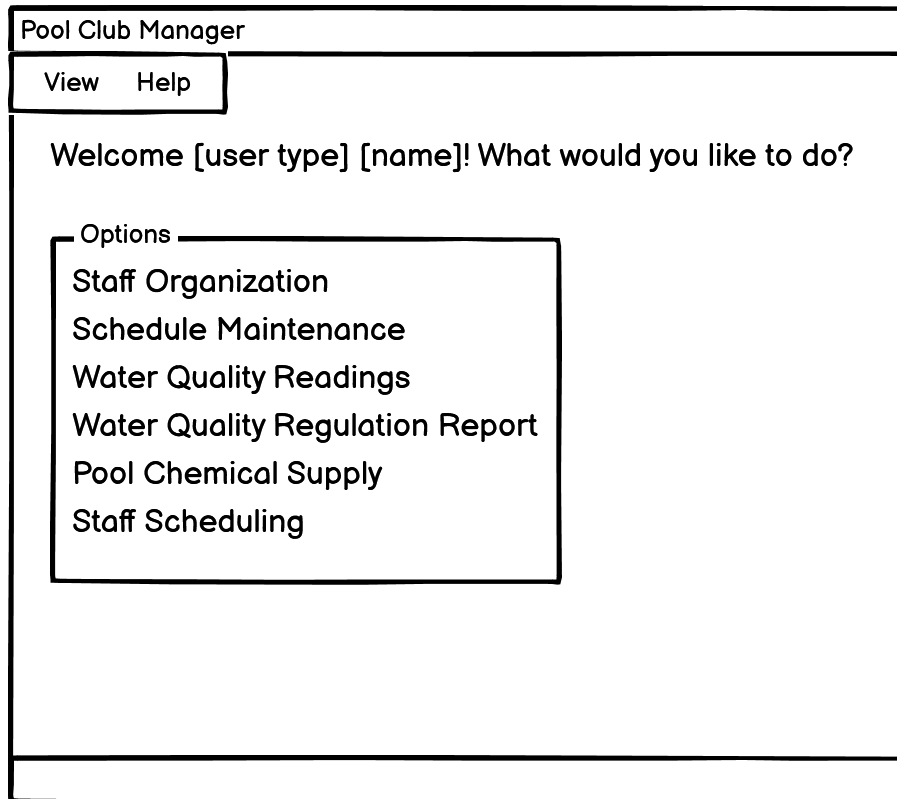


Figure 2 – Wireframe for Home screen

Pool Club Manager

File

Edit

View

Help

[Home](#) > [Staff Organization](#) > All Staff Members

ID_num	FirstName	LastName	HireDate	SeparationDate	ReasonForSeparation	Passwords	UserType
001	Samson	Cloud	08/23/1998		N/A	*****	Pool Operator/Lifeguard
002	Candy	Bar	08/23/1999		N/A	*****	Lifeguard
003	Clark	Bar	08/23/1999		N/A	*****	Lifeguard
004	Hershee	Kiss	08/23/2000		N/A	*****	Lifeguard
005	Pack	Mule	08/23/2000		N/A	****	Employee
006	Janet	Torr	08/23/2000		N/A	****	Employee
007	Phi	Nancer	08/23/2000		N/A	****	Employee

Record

ID num

First Name

Last Name

Hire Date

Separation Date

Reason for Separation

Password

User Type

Actions

Add Record

Update Record

Remove Record

Remove Multiple Records

Clear Entry Fields

Figure 3 – Wireframe for Staff Member screen

Pool Club Manager

File Edit View Help

[Home](#) > Schedule Maintenance

ItemNumber	Date	Task	StaffRep.	Status
000				
001				
002				
003				
004				
005				
006				

Record

ItemNumber Date Staff Rep. Status

Task

Actions

Figure 4 – Wireframe for Schedule Maintenance screen

Pool Club Manager

File Edit View Help

[Home](#) > [Water Quality Readings](#) > Daily Readings

Day_ID	Date	Time	pH	Total Chlorine	Free Chlorine	Combined Chlorine
0000						
0001						
0002						
0003						
0004						
0005						
0006						

Record

Day_ID Date Time

pH Total Chlorine Free Chlorine Combined Chlorine

Actions

Figure 5 – Wireframe for Daily Readings screen

Pool Club Manager

File Edit View Help

[Home](#) > [Water Quality Readings](#) > Weekly Readings

Week_ID	Date	Time	Total Hardness	Total Alkalinity	CYA (stabilizer)
0000					
0001					
0002					
0003					
0004					
0005					
0006					

Record

Week_ID Date Time

Total Hardness Total Alkalinity CYA (stabilizer)

Actions

Figure 6 – Wireframe for Weekly Readings screen

Pool Club Manager

File Edit View Help

[Home](#) > Pool Chemical Supply

Chemical Name	Amount Purchased	Date Purchased	Amount Used	Date Used	Amount On-hand
Chlorine					
Calcium chloride					
Sodium bicarbonate					
Sodium bisulfate					
Sodium carbonate					
Sodium hypochlorite					
CYA					

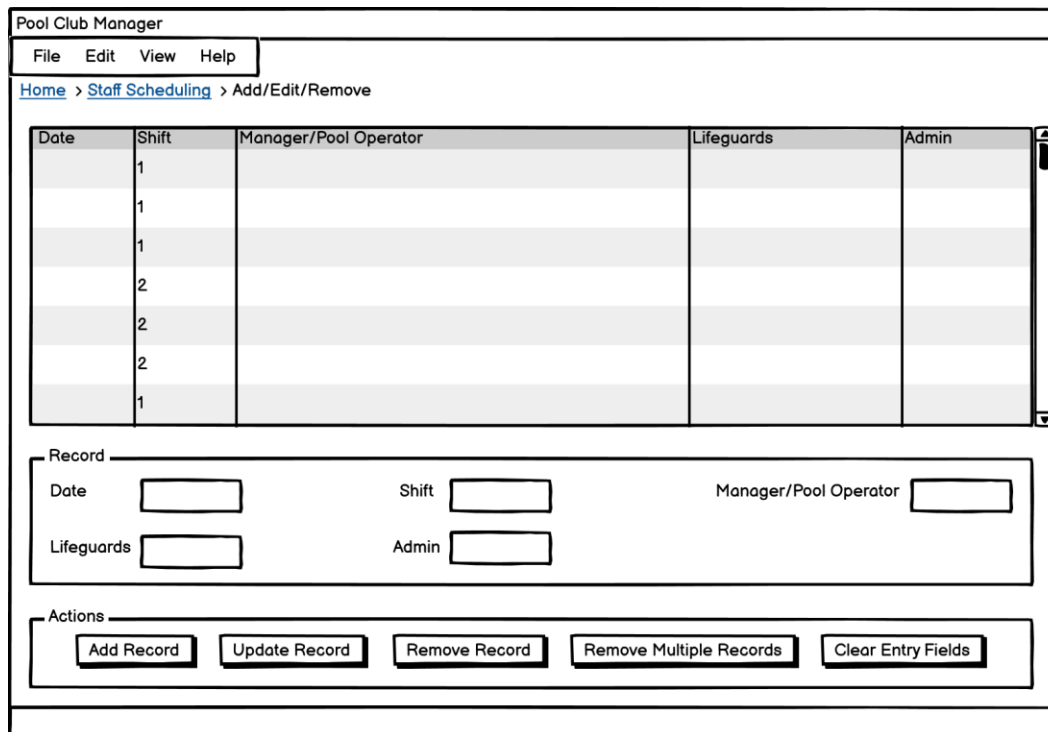
Record

Chemical Name Amount Purchased Date Purchased

Amount Used Date Used Amount On-hand

Actions

Figure 7 – Wireframe for Pool Chemical Supply screen



The wireframe for the Staff Scheduling screen is titled "Pool Club Manager". It features a menu bar with "File", "Edit", "View", and "Help". Below the menu is a breadcrumb trail: "Home > Staff Scheduling > Add/Edit/Remove". The main area contains a table with five columns: "Date", "Shift", "Manager/Pool Operator", "Lifeguards", and "Admin". The table has 10 rows, with the first column containing dates (1, 1, 1, 2, 2, 2, 2, 1) and the second column containing shift numbers (1, 1, 1, 2, 2, 2, 2, 1). Below the table is a "Record" section with input fields for "Date", "Shift", "Manager/Pool Operator", "Lifeguards", and "Admin". At the bottom is an "Actions" section with buttons for "Add Record", "Update Record", "Remove Record", "Remove Multiple Records", and "Clear Entry Fields".

Date	Shift	Manager/Pool Operator	Lifeguards	Admin
1	1			
1	1			
1	1			
2	2			
2	2			
2	2			
2	2			
1	1			

Record

Date Shift Manager/Pool Operator

Lifeguards Admin

Actions

Figure 8 – Wireframe for Staff Scheduling screen

III-c. Use Cases

Shown below are the use case diagrams from the planning phase. Each diagram identifies the intended user and the functions they will have access to based on their type of account. For reference, here are the use cases covered below:

- Figure 9 is the Pool Manager's account.
- Figure 10 is the Pool Operator's (who is also a Lifeguard) account.
- Figure 11 is a Lifeguard account.
- Figure 12 is a general-purpose Employee account.

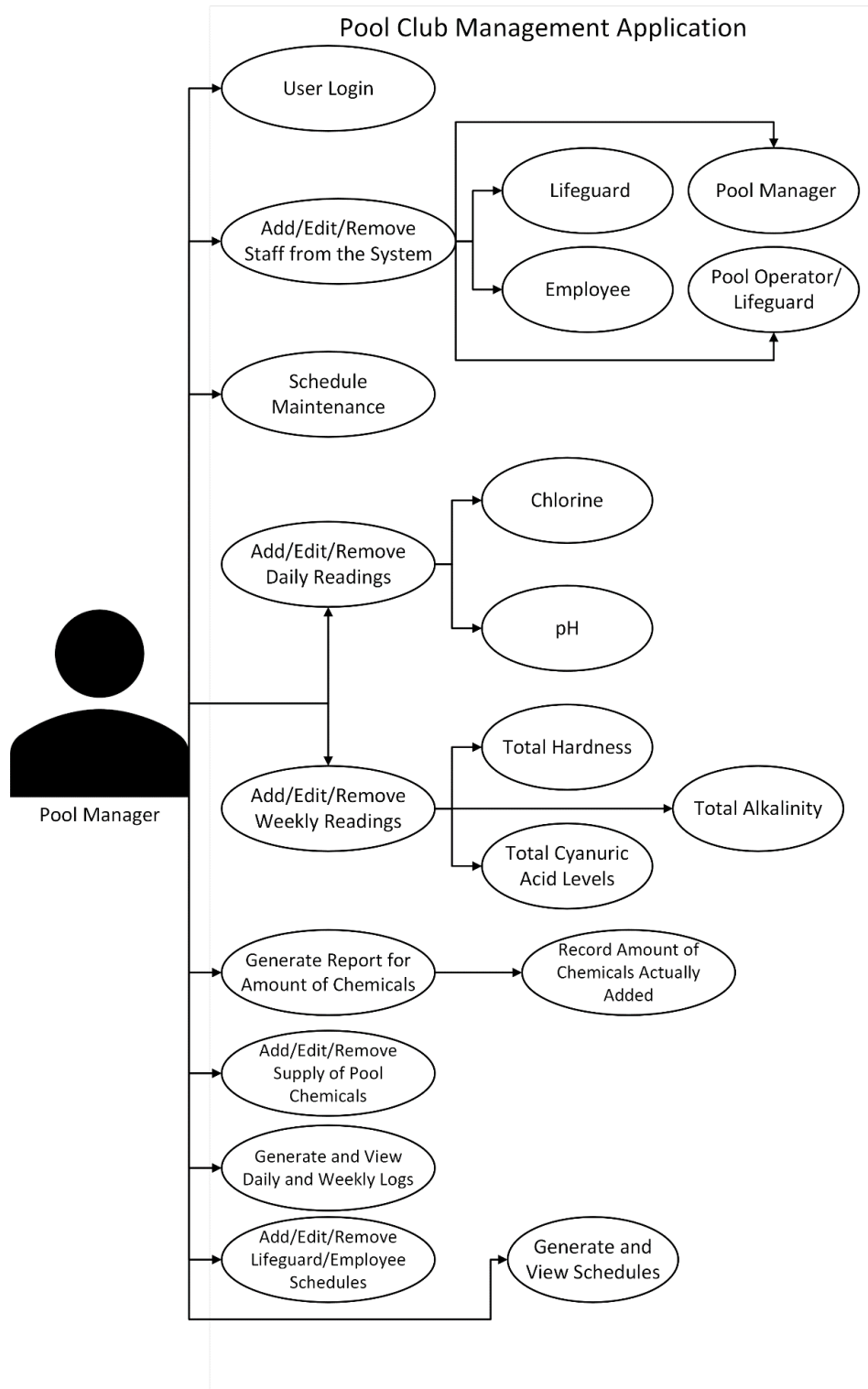


Figure 9 – Use Case Diagram for Pool Manager

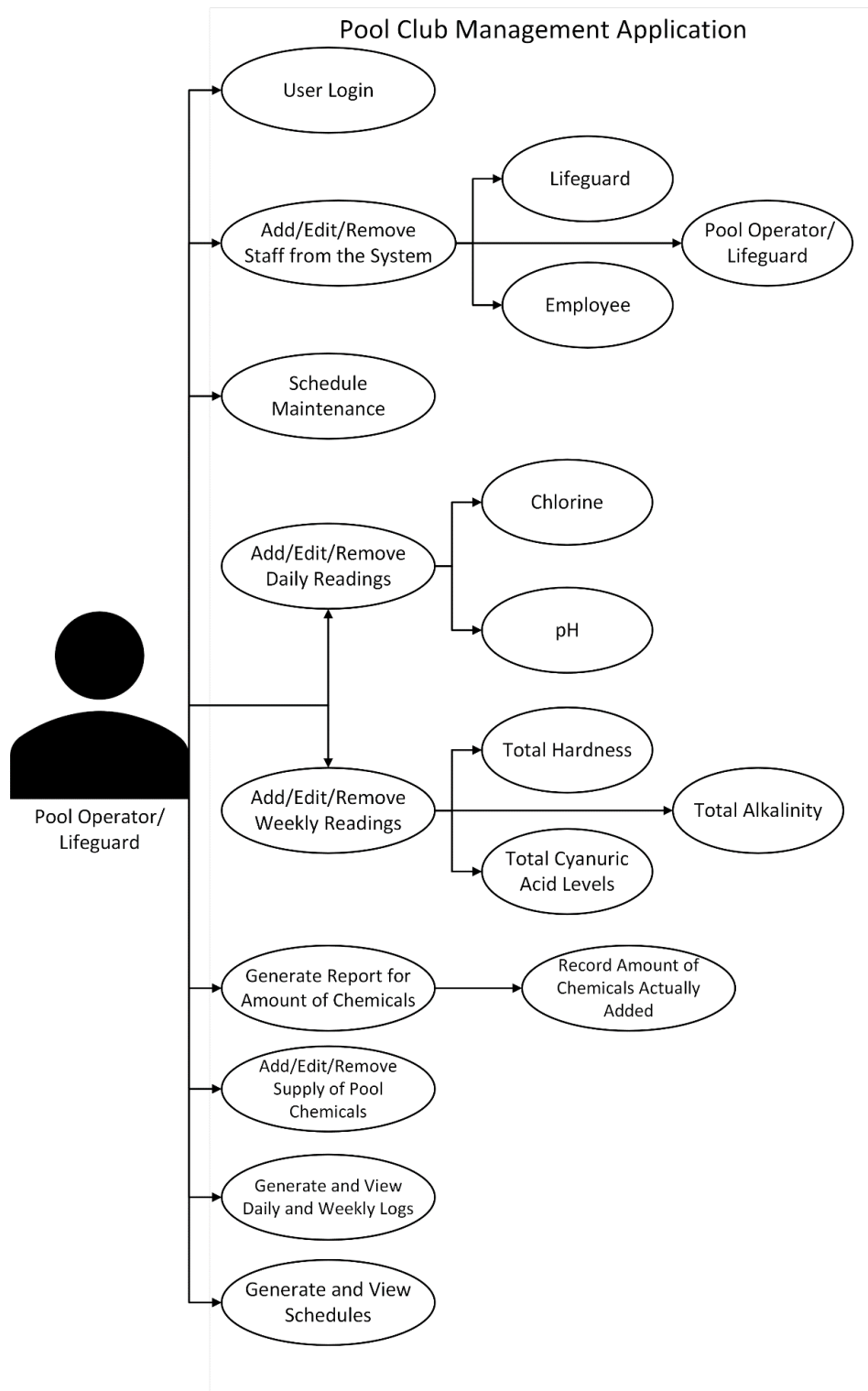


Figure 10 – Use Case Diagram for Pool Operator/Lifeguard

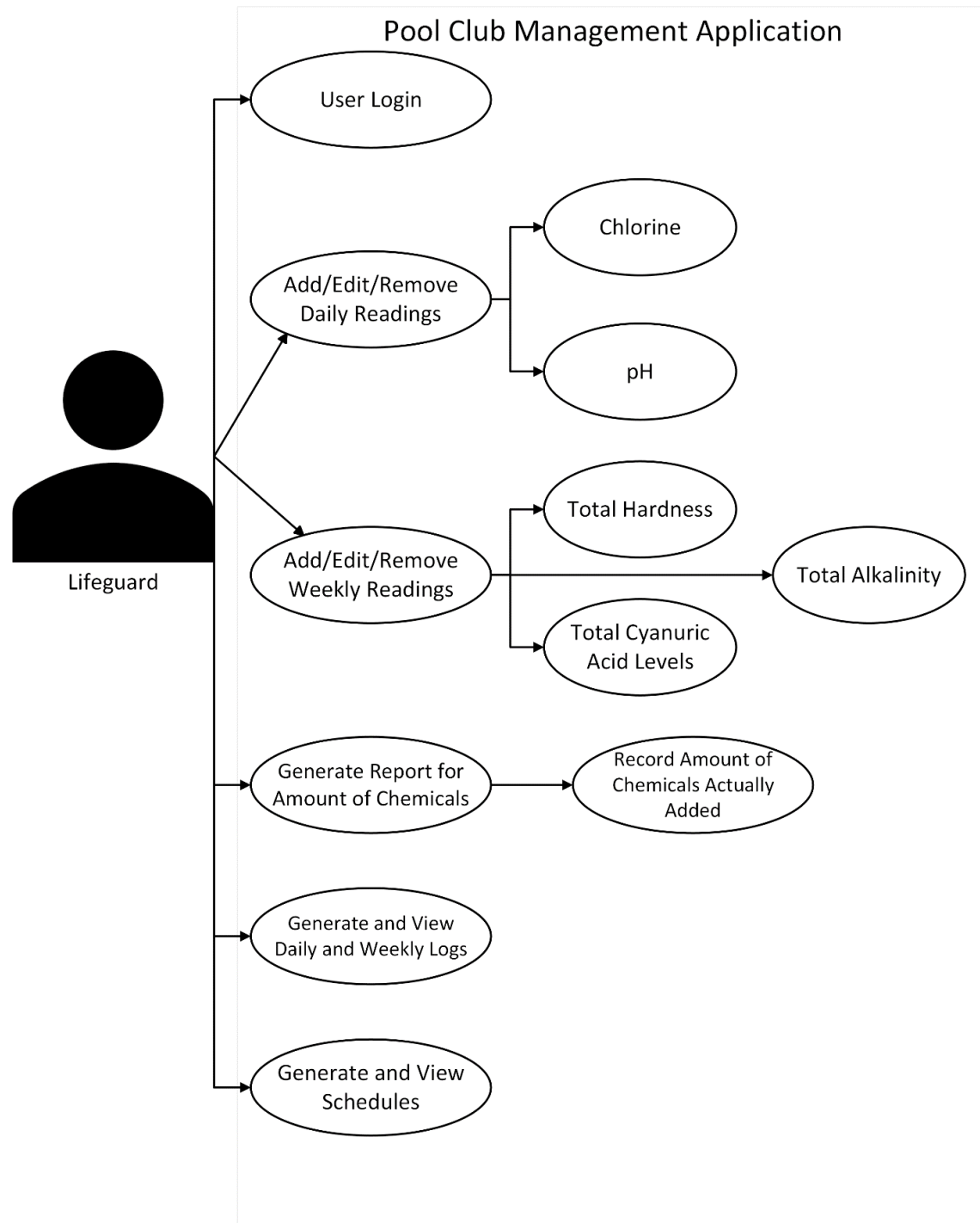


Figure 11 – Use Case Diagram for Lifeguard

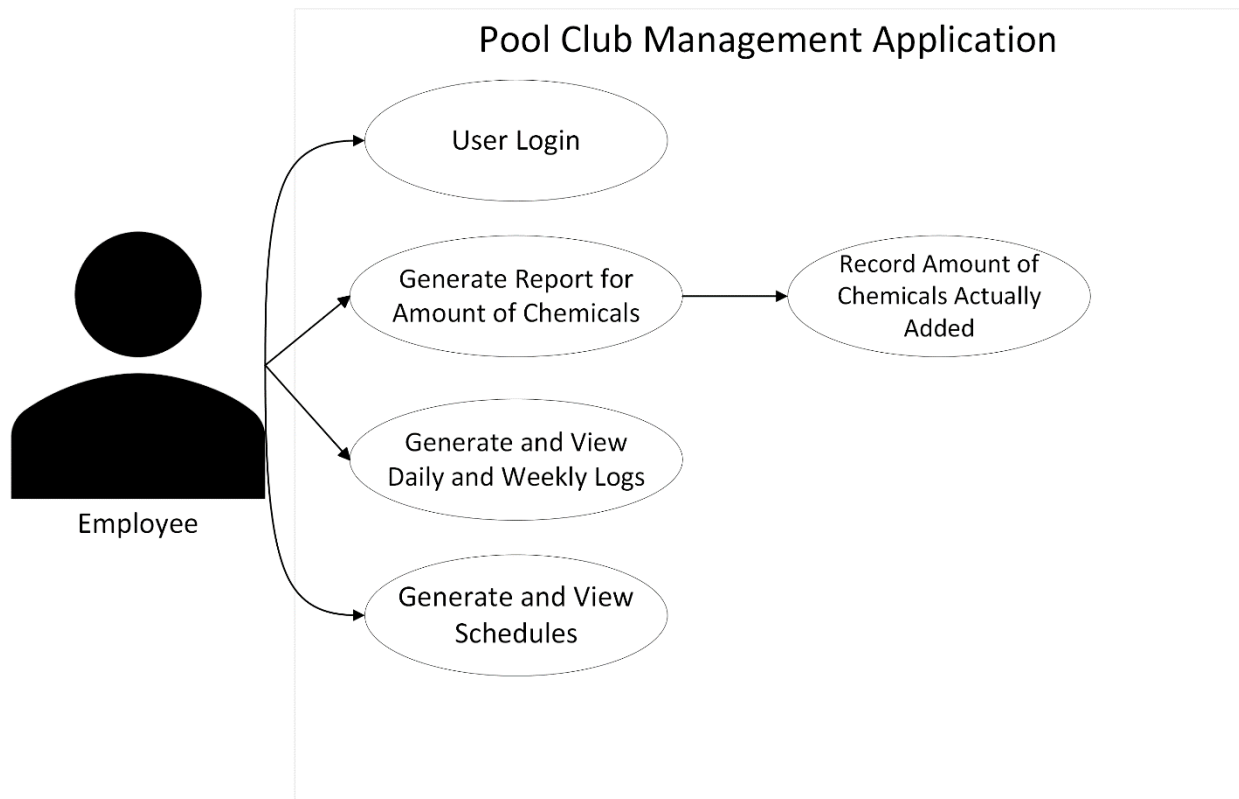


Figure 12 – Use Case Diagram for Employee

IV. Functionality

The diagram below shows how the system functions at the hardware level.

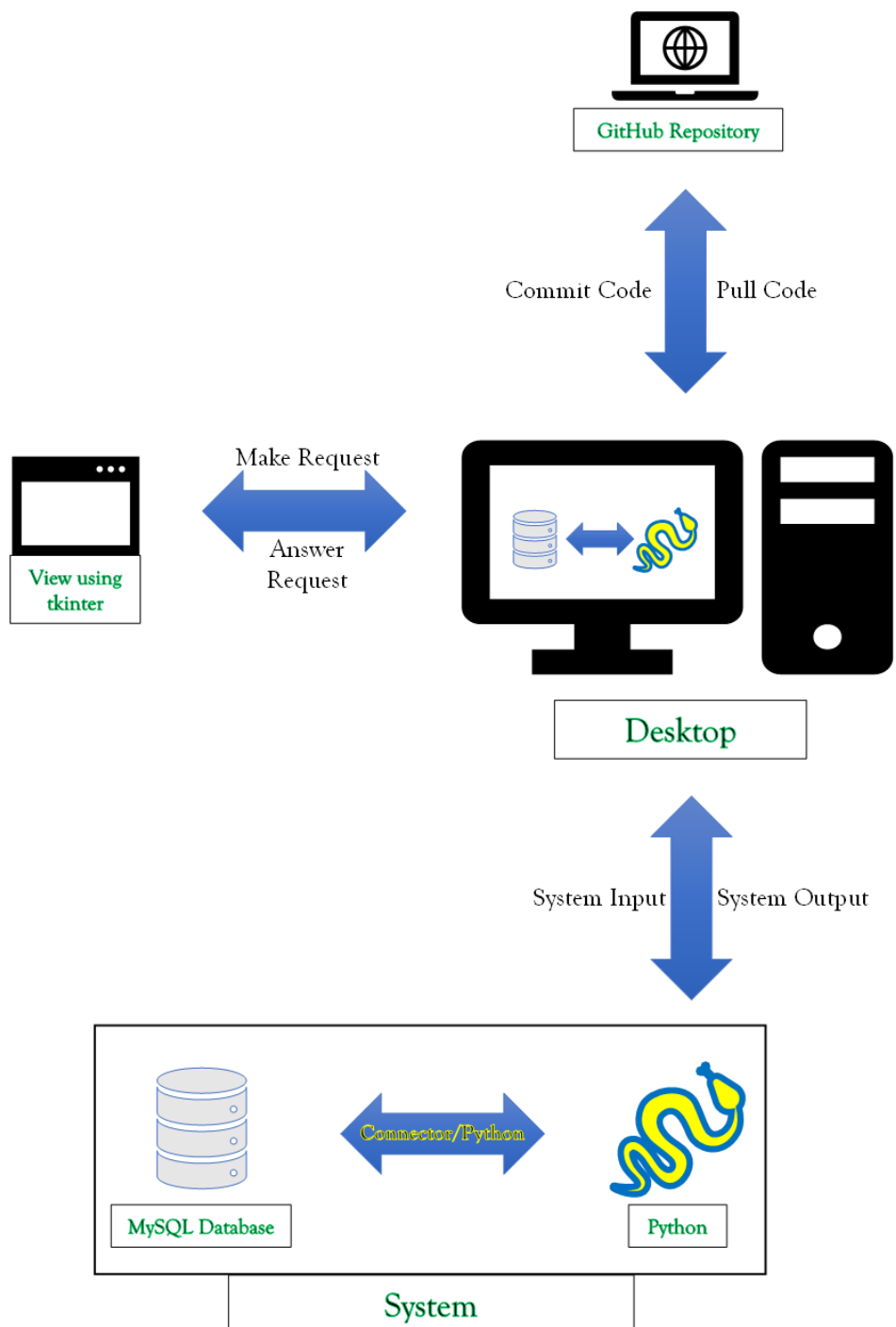


Figure 13 – Preliminary Architectural Plan and the current system layout

This software is made for use on a desktop computer. The GUI files, the API files, and the MySQL database are all stored on the same desktop. The tkinter GUI provides a mechanism in which to request information from the system and display information in response to the request. Python is used to code the APIs which perform the functions identified in the use case scenarios. MySQL stores the data for future retrieval and analysis. Last but not least, MySQL Connector/Python is used to allow Python and MySQL to communicate with each other.

From a software perspective, the interface is pretty simple. You have a menu with buttons that can take you to six different screens: Staff Organization, Schedule Maintenance, Daily Readings, Weekly Readings, Pool Chemical Supply, and Staff Scheduling.

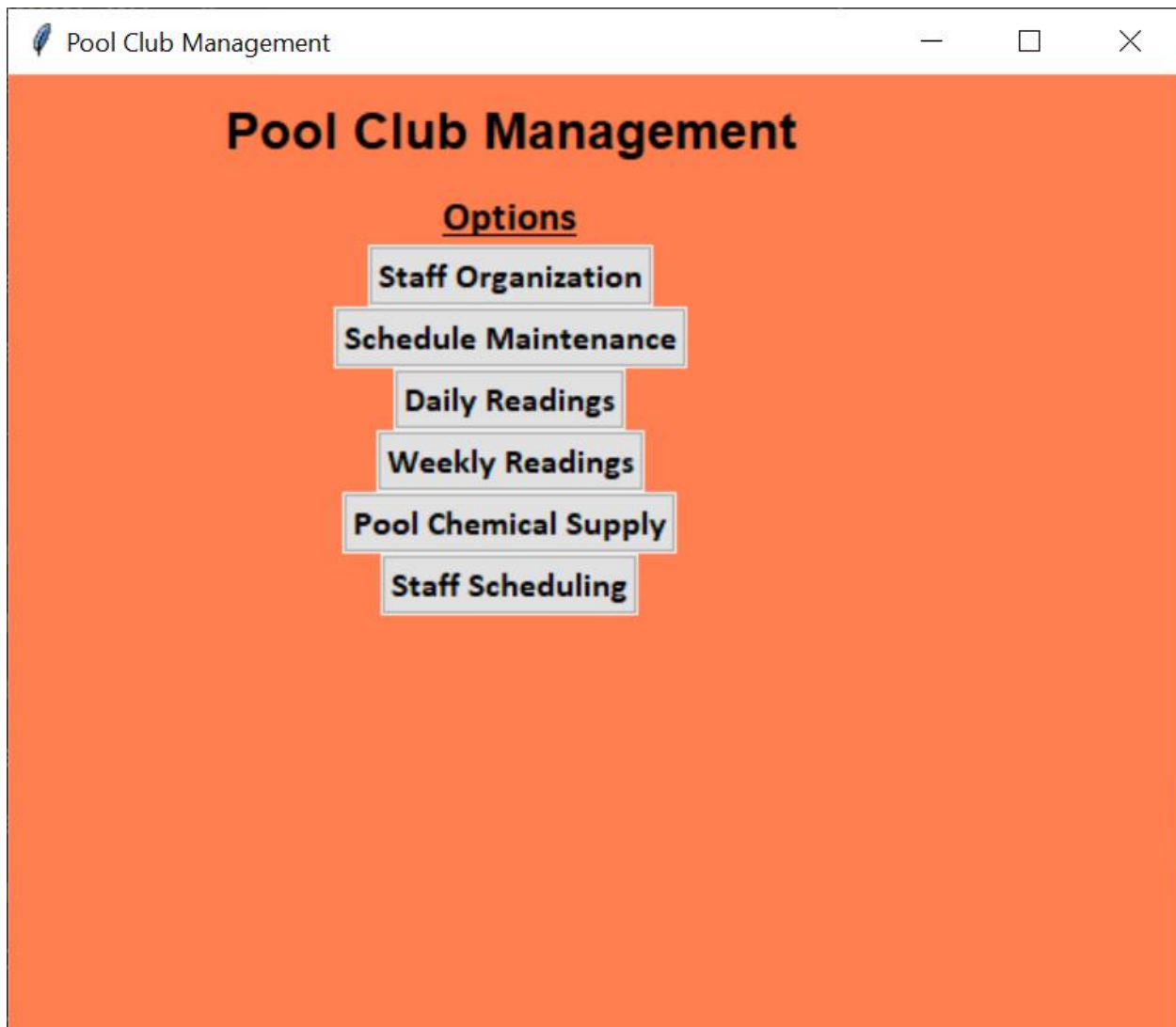


Figure 14 – The Pool Club Management Software’s home screen.

On each screen, there is a treeview that shows the data from the database, a Record frame to input parameters for the different records, and a button frame to perform actions on the records based on what is currently present within the fields of the Record frame.

ID Num	First Name	Last Name	Hire Date	Separation Date	Reason for Separation	Password	User Type
1	John	Smith	1998-06-06	2014-07-11	Retirement	T1m3@ut	Pool Manager
2	Samson	Cloud	1998-08-23		N/A	Oper8er	Pool Operator/Lifeguard
3	Candy	Bar	1999-08-23		N/A	6u@rd	Lifeguard
4	Clark	Bar	1999-08-23		N/A	6u@rd	Lifeguard
5	Hershee	Kiss	2000-08-23		N/A	6u@rd	Lifeguard
6	Pack	Mule	2000-08-23		N/A	W0rk	Employee
7	Janet	Torr	2000-08-23		N/A	W0rk	Employee
8	Phi	Nancer	2000-08-23		N/A	W0rk	Employee
9	Jane	Doe	2006-06-06		N/A	T1m3@ut	Pool Manager
10	Honey	Cheeryo	2007-05-15		N/A	6u@rd	Lifeguard

Record

ID Num First Name Last Name

Hire Date Separation Date Reason for Separation

Password User Type

Actions

Figure 15 – The Pool Club Management Software’s Staff Organization screen.

Pool Club Manager - Schedule Maintenance

Schedule Maintenance

Item Number	Date	Task	Staff Representative	Status
1	2023-11-18	Add chlorine	2	Complete
3	2023-11-18	Take weekly readings	2	Complete

Record

Item Number Date Task

Staff Rep. Status

Actions

Figure 16 – The Pool Club Management Software's Schedule Maintenance screen.

Pool Club Manager - Daily Readings

Daily Readings

Day ID	Date	Time	pH	Total Chlorine	Free Chlorine	Combined Chlorine
1	2023-11-07	11:00:00	7.0	3.0	3.0	0.0
2	2023-11-07	13:00:00	7.0	4.0	4.0	0.0
3	2023-11-07	14:00:00	7.0	4.0	4.0	0.0
4	2023-11-08	11:00:00	7.2	4.2	4.1	0.1
5	2023-11-08	12:00:00	7.1	5.0	4.8	0.2

Record

Day ID Date Time

pH Total Chlorine Free Chlorine Combined Chlorine

Actions

Figure 17 – The Pool Club Management Software’s Daily Readings screen.

Pool Club Manager - Weekly Readings

Weekly Readings

Week ID	Date	Time	Total Hardness	Total Alkalinity	CYA (stabilizer)
1	2023-11-07	17:00:00	178	100	81
2	2023-11-13	17:00:00	172	68	60
3	2023-11-20	17:00:00	210	113	76
4	2023-11-27	17:00:00	185	62	85

Record

Week ID Date Time

Total Hardness Total Alkalinity CYA (stabilizer)

Actions

Figure 18 – The Pool Club Management Software’s Weekly Readings screen

Pool Club Manager - Pool Chemical Supply

Pool Chemical Supply

Chemical ID	Chemical Name	Amount Purchased	Date Purchased	Amount Used	Date Used	Amount On-hand
1	Chlorine	3	2023-11-07	0	2023-11-13	15
2	Calcium chloride	3	2023-11-07	0	2023-11-13	15
3	Sodium bicarbon	None	None	None	None	None
4	Sodium bisulfate	None	None	None	None	None
5	Sodium carbonat	None	None	None	None	None
6	Sodium hypochlc	None	None	None	None	None
7	CYA	None	None	None	None	None

Record

Chemical ID Chemical Name Amount Purchased

Date Purchased Amount Used Date Used

Amount On-hand

Actions

Figure 19 – The Pool Club Management Software’s Pool Chemical Supply screen.

Schedule ID	Date	Shift	Manager/Pool Operator	Lifeguards	Admin
2	2023-11-18	2	2	10	2
3	2023-11-18	1	2	12	2

Record

Schedule ID Date Shift

Manager/Pool Operator Lifeguards Admin

Actions

Figure 20 – The Pool Club Management Software’s Staff Scheduling screen.

For more information, please consult the manual.

V. Challenges

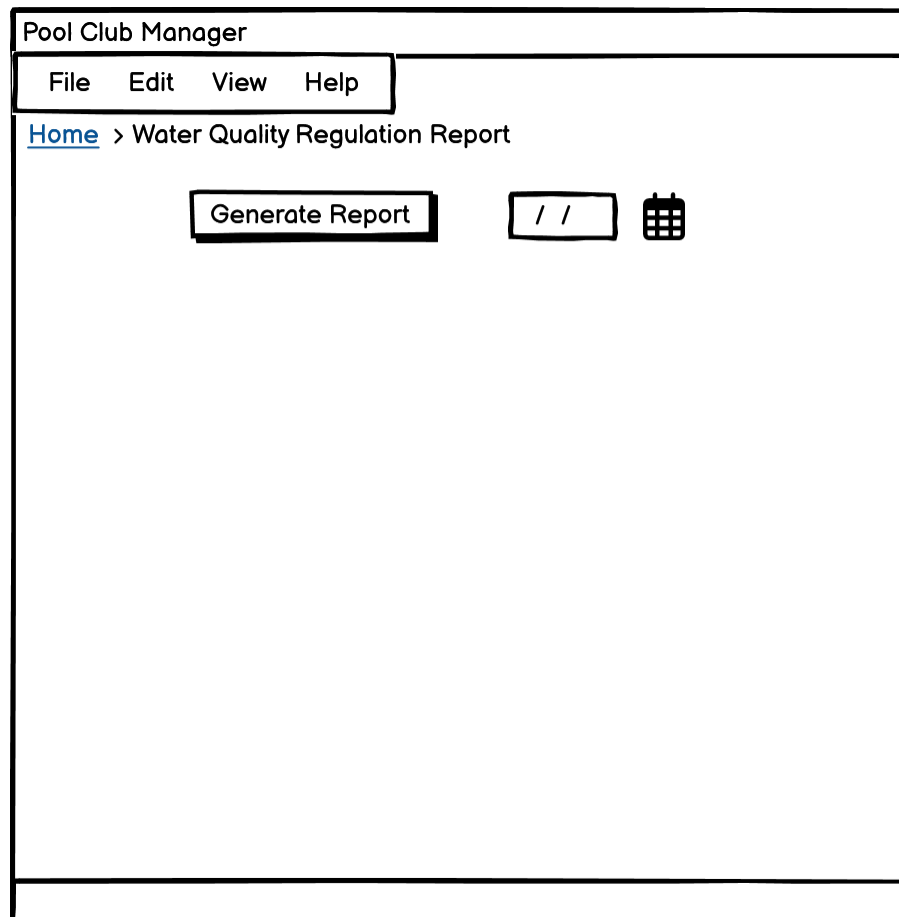
There were a couple of challenges along the way that I faced with courage. My problems can be broken up into two big categories: programming limitations and time constraints.

V-a. Programming Limitations

There have been a couple of programming limitations that have impeded the progress of my original vision for this project.

One of these limitations was learning how to use Python. This challenge was self-imposed since I have had very limited experience with Python up until now (with tkinter experience being non-existent). I decided to pick up this project in part to learn this language to increase my marketability for job applications. Considering that I was near beginner-level with this language, I believe that I have improved by a decent margin.

Another challenge was how it was originally planned for the Water Quality Regulation Reports (both in regard to reporting the Daily Readings and reporting the Weekly Readings) to show a preview of the PDF document. After entering a date into an entry field, you would get to view the document before either downloading it as a file or being given the option to print the document right away. Shown below are wireframe mock-ups of what the feature would have looked like.



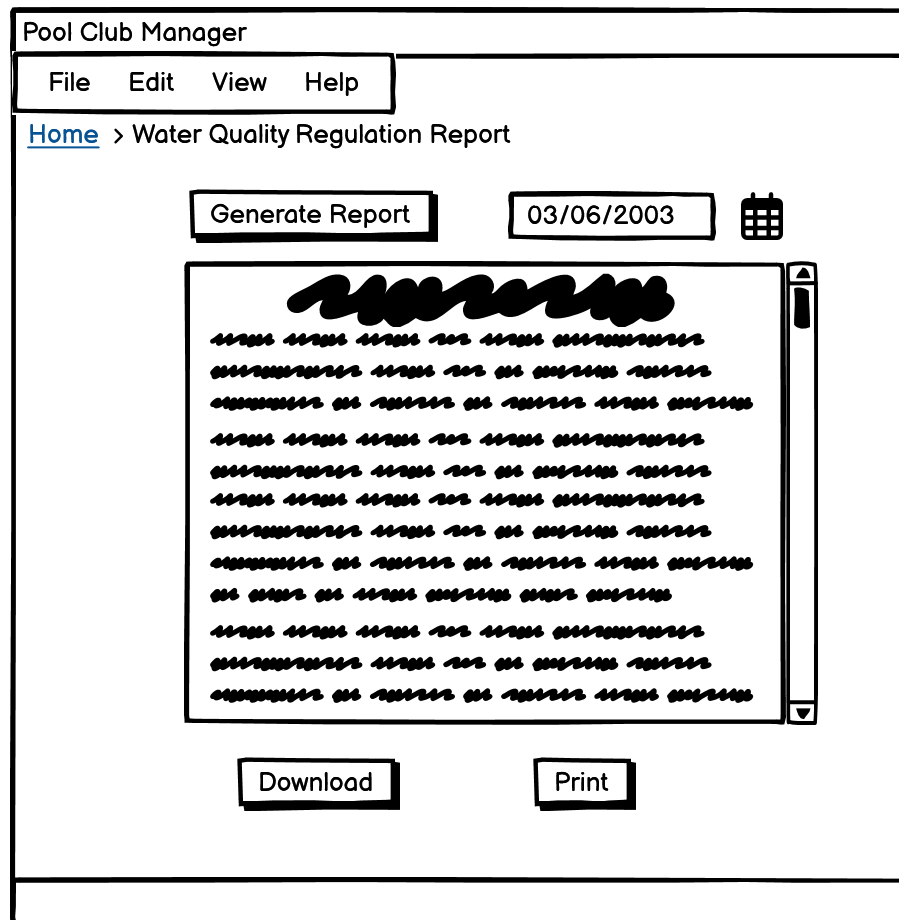


Figure 21 – Wireframes of the planned Water Quality Regulation Report screen, before and after input of the date.

While I was able to generate the PDF document, displaying a preview of the document in a GUI environment before downloading it seemed to be impossible. Thus, I had to relegate the creation and downloading of the Water Quality Regulation Reports to additional buttons in both the Daily Readings and Weekly Readings screens.

V-b. Time Constraints

There were a few features I couldn't implement due to the time constraints. In hindsight, all of the features I originally planned to do would have required more time than allotted in a semester. The first feature I could not implement was the login screen, which is pictured below.

The wireframe shows a window titled "Pool Club Manager". Inside the window, there is a menu bar with "View" and "Help" options. The main area contains a login form with the following elements:

- A label "Username:" followed by a text input field.
- A label "Password:" followed by a text input field.
- A "Login" button centered below the input fields.

The window has a standard title bar and a footer bar.

Figure 22 – Wireframe of the planned Login screen

In order to make sure the program worked, I decided to save the user login feature for last. Unfortunately, as a result, I ran out of time to implement this feature as this was more than a simple login screen: it was meant to show different views of the same screen based on your account's user type. For comparison, the first mockup below would be what Pool Manager and Pool Operator/Lifeguard accounts would be able to access, while the mock up directly following it was the planned menu for Lifeguard and Employee accounts.

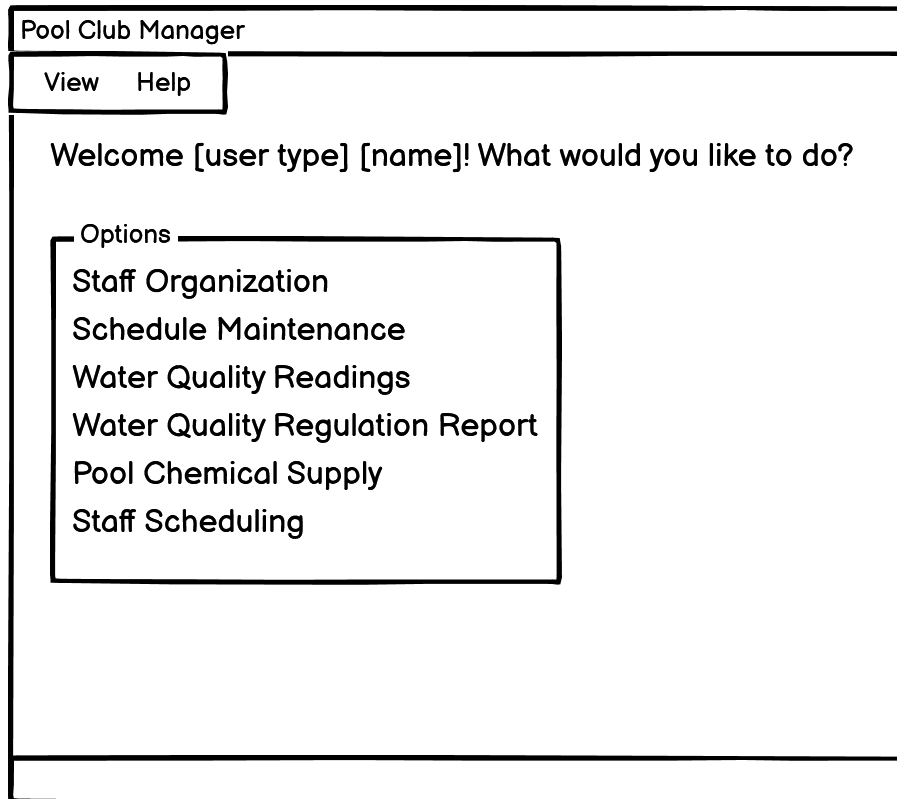


Figure 23 – Wireframe of the Home screen for both Pool Manager and Pool Operator/Lifeguard accounts.

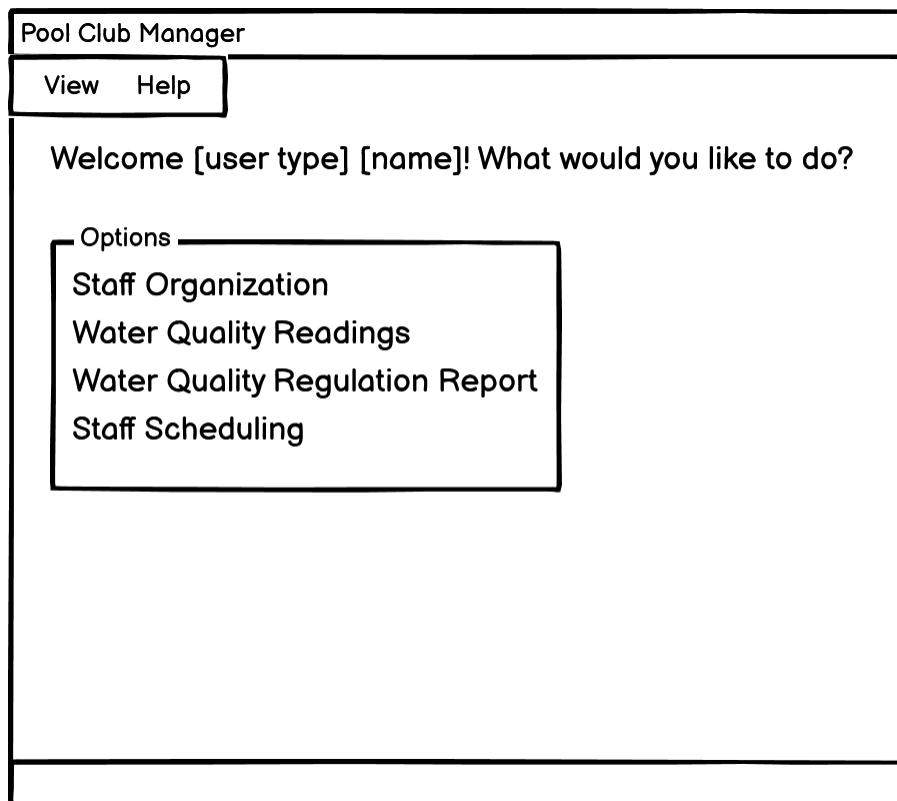


Figure 24 – Wireframe of the Home screen for both Lifeguard and Employee accounts.

Last but not least, I wish I had the time to figure out how to sort and filter what is displayed on the treeview. Basically, I wanted different ways of showing data on the screens in order to make finding the desired records easier to locate. Using SQL commands to sort data was doable, but coming up with an easy method of filtering by specific fields and then displaying it in a GUI proved to be more difficult than I had envisioned.

VI. What I Wish I Knew at the Beginning of the Project

One thing I wish I knew at the beginning of the project was how APIs interact with a graphical user interface. I wouldn't have had to rewrite the API files then. Before preparing for this project, I have not worked with the concept of APIs before. My previous experience with coding was having all of the functions within one executable file. When I was first developing the APIs, I was including an input interface using the console. When it came to integrating a GUI, I essentially had to rewrite the API files so it can be used by any other input mechanism.

Another point of interest that I wish to share is how I wish I found a simpler way to input the variables into an SQL command without having to overwrite every variable with what is written (or not written) in the fields of the Record frame in each screen.

VII. Future Plans for the Pool Club Management Software

My future plans for the Pool Club Management Software are as follows:

- Login screen
- Different account permissions
- Expand functionality to match that of states outside of Maryland.

For the login screen, I will not only have a username and password, but also implement two-factor authentication. Once logged in, the screens and actions shown to the user will differ depending on the User Type. Another feature that will be implemented is the ability to sort and filter the records currently displayed on the treeview of each screen.

As I only have information on Maryland rules and regulations, I was only able to create the software based off of those specifications. In future iterations of this software, I plan for it to also account for other state and regulations for USA pool clubs.

Once all the tasks listed above are complete, I plan on monetizing this project and adding it to my portfolio. This will help in landing future job opportunities.

VIII. Conclusion

In conclusion, I have created a functional user interface for the Pool Club Management Software using Python and MySQL. This is my first major solo project that proved to be a good and very satisfying challenge. It has involved the use of skills I have learned throughout all of my AIT courses. Additionally, I learned a lot of new software such as Balsamiq and Python tkinter as well as expanding my knowledge of other software such as Python and MySQL. I also learned how to successfully connect a program coded in the Python language to a MySQL database while also utilizing MySQL syntax.

Although I was not able to implement all of the functions I originally identified in the proposal, I did learn how to compartmentalize a project so that I can complete portions of the project so that it ends up functional. This was also the first time I have used APIs on a project. Although there was a learning curve, I now understand how an API functions and can connect to different user interfaces.

I plan to continue developing this software and continue developing my programming skills. This has been a great learning experience that will help with future endeavors.

IX. Acknowledgements

There are several people and resources I would like to thank for helping me get this far. I shall start by thanking the people who have helped make this project possible. First, I would like to thank my advisor Prof. Jal Irani. He was both patient with me and helped me to the best of his ability, even going so far as learning some Python to help with coding problems. Next, I would like to thank the AIT department for the classes I previously took and giving me the foundation to be able to construct the deliverables for this project. I am thankful to the AIT Program Director Cheryl Brown for approving my proposal for this case study. I would also like to thank my family—my mother and father—for keeping me on the right track and reviewing some of my material for readability.

In terms of resources that have helped me throughout this project, I have LinkedIn Learning and freeCodeCamp.org to be thankful for. LinkedIn Learning helped with learning the Python tkinter technology, which is what allows for a Python-based program to construct and utilize a GUI. freeCodeCamp.org is a volunteer-based coding resource that helped me learn about Python's version of case-switch, which helped in the development of the pre-GUI versions of the APIs instead of having a few faulty if-else statements.

References

- [1] GitHub, “Let’s build from here,” GitHub, [Online], <https://github.com/> (accessed Jun. 27, 2023).
- [2] L. Barbosa and A. Hora, “How and Why Developers Migrate Python Tests,” in 2022 IEEE International Conference on Software Analysis, evolution and reengineering (saner), 2022, pp. 538–548
- [3] MySQL.com, “MySQL Connector/Python Developer Guide,” MySQL, [Online], <https://dev.mysql.com/doc/connector-python/en/> (accessed Jun. 28, 2023).
- [4] Python Software Foundation, “Python 3.11.4 documentation,” 3.11.4 Documentation, [Online], <https://docs.python.org/3/> (accessed Jun. 28, 2023).
- [5] Python Software Foundation, “tkinter - Python interface to TCL/TK,” Python documentation, [Online], <https://docs.python.org/3/library/tkinter.html> (accessed Jun. 28, 2023).
- [6] Python Software Foundation, “unittest — Unit testing framework,” Python documentation, [Online], <https://docs.python.org/3/library/unittest.html> (accessed Jun. 27, 2023).
- [7] M. Reichardt, M. Gundall, and H. D. Schotten, “Benchmarking the Operation Times of NoSQL and MySQL Databases for Python Clients,” in IECON 2021 - 47th annual conference of the IEEE industrial electronics society, 2021
- [8] B. A. Meier, Python Gui Programming Cookbook: Develop Beautiful and Powerful Guis Using the Python Programming Language, Second. Birmingham, UK: Packt Publishing, 2017.
- [9] Chris, K. (2023, March 28). Python switch statement – switch case example. freeCodeCamp.org. <https://www.freecodecamp.org/news/python-switch-statement-switch-case-example/> (accessed Oct. 15, 2023)
- [10] Stone, B. (2018). Create a GUI for your Python desktop application - Python GUI Development with Tkinter Video Tutorial: LinkedIn Learning, formerly Lynda.com. LinkedIn. LinkedIn Learning. Retrieved August 8, 2023, from <https://www.linkedin.com/learning/python-gui-development-with-tkinter-2/create-a-gui-for-your-python-desktop-application?u=56674649>.

Appendix A – Proposal

Proposal Date: 08/18/2023**AIT CASE STUDY COVER SHEET**Student Name + TU ID: Evan Lyle (ID: 0619167 515)Semester: Fall 2023Case Study Title: Pool Club Management ApplicationAdvisor Name: Prof. Jal IraniAbstract:

Do you know how disorganized pool clubs really are? Pool operations are often mishandled, whether due to staff shortages, staff turnover, underspending or overspending on resources like chemicals, being unprepared for local health inspections, or even simply losing track of metrics such as the pH level or alkalinity. Such behavior needs to be rectified. This project will focus on developing an application for managing a pool club database with a functional and user-friendly interface. The resulting application will allow you to add, edit, and remove various chemicals and staff members (system users, lifeguards, and other employees); schedule different duties among the staff members; and monitor and track the water quality in the pool. It will also track the amount of pool chemicals in-stock.

The main programming language will be Python, with Python tkinter being used for the GUI, Python unittest in order to test the Python code, and GitHub being used for configuration management in order to store different versions of the code. The database itself will be written and stored in MySQL, with the MySQL Connector/Python tool allowing for the Python interface to interact with MySQL.

Course:

- ☐ AIT 710
- ☐ AIT 715
- ☐ AIT 720
- ☒ AIT 725
- ☐ AIT 730

Past Courses: Please list all course completed and currently taking. Remember that you must complete all prerequisites in order to register for a case study.

<i>Number</i>	<i>Name</i>	<i>Semester</i>	<i>Grade</i>
AIT 600	Information Technology Infrastructure	Fall 2020	A
AIT 610	Systems Development Process	Fall 2020	A-
AIT 616	FUND WEB TECH/DEVELOPMENT	Spring 2021	A
AIT 624	Software Engineering Fundamentals	Spring 2021	B+
AIT 613	Introduction to Software Security	Fall 2021	A-
AIT 618	Client/Server-side Programming on the Web	Fall 2021	A
AIT 620	Business Data Communications	Spring 2022	A
AIT 632	Database Management Systems	Spring 2022	A
AIT 641	Software Requirements Engineering	Fall 2022	A
AIT 642	Software Testing and Maintenance	Spring 2023	B+

FOR OFFICE USE ONLY

Proposal Approved:

Report Received:

Pool Club Management Application

Problem Description

There is a need to develop an application to assist in swimming pool club management using modern technologies such as Python and MySQL. Too many pool clubs around the country have been plagued with mismanagement, causing irregular hours for pool staff and the management of chemicals (like the pH levels) to be ignored. Many pool clubs exceed their budget due to overusing pool chemicals, running short of pool chemicals, and mismanaging lifeguard scheduling. Additionally, pools are frequently inspected by local health departments. Many times, pools are not prepared for these inspections due to incomplete or missing documentation concerning water testing and licensing certifications resulting in temporary closure. The goal of this project is to create a database using modern technologies to keep track of complex pool tasks, water quality, maintenance, documentation, and scheduling so that financial and staff availability concerns will become a thing of the past.

The objective of this project is to provide an application that helps to keep track of your resources. For example, monitoring and tracking levels of chlorine and pH on a daily basis and total hardness, total alkalinity, and cyanuric acid levels on a weekly basis. For pool maintenance these include the amount of chemicals used versus the amount of chemicals that are in-stock; the ability to track and input the amount and type of chemicals recently purchased/stocked; and reminders of which chemicals need to be restocked based on scheduled use and if chemicals fall

below a certain threshold. Another function of the database is the ability to add, edit, delete, and schedule for lifeguards and employees to oversee and manage the pool.

By the end of this project, the following deliverables will be presented:

- Requirements documentation,
- Database ER diagram,
- The use case diagram,
- The user interface diagram,
- The application's source code,
- A user manual,
- The scripts and tools for creating the project, and
- The final presentation and its recording.

Justification of the problem

I plan to use all the skills that I have learned across my AIT courses as it relates to software engineering. These include my experience with learning how to create the appropriate documentation for understanding the system at different levels of development, inputting, organizing, and editing the data in the proper format, and how to connect a database to a GUI interface. I shall also demonstrate the use of GitHub, showing how I can keep track of different versions of the project. I believe this project is worthy of the case study since all of those skills demonstrate how technology is a great asset for keeping track of resources and scheduling as long as it is planned out and constructed properly.

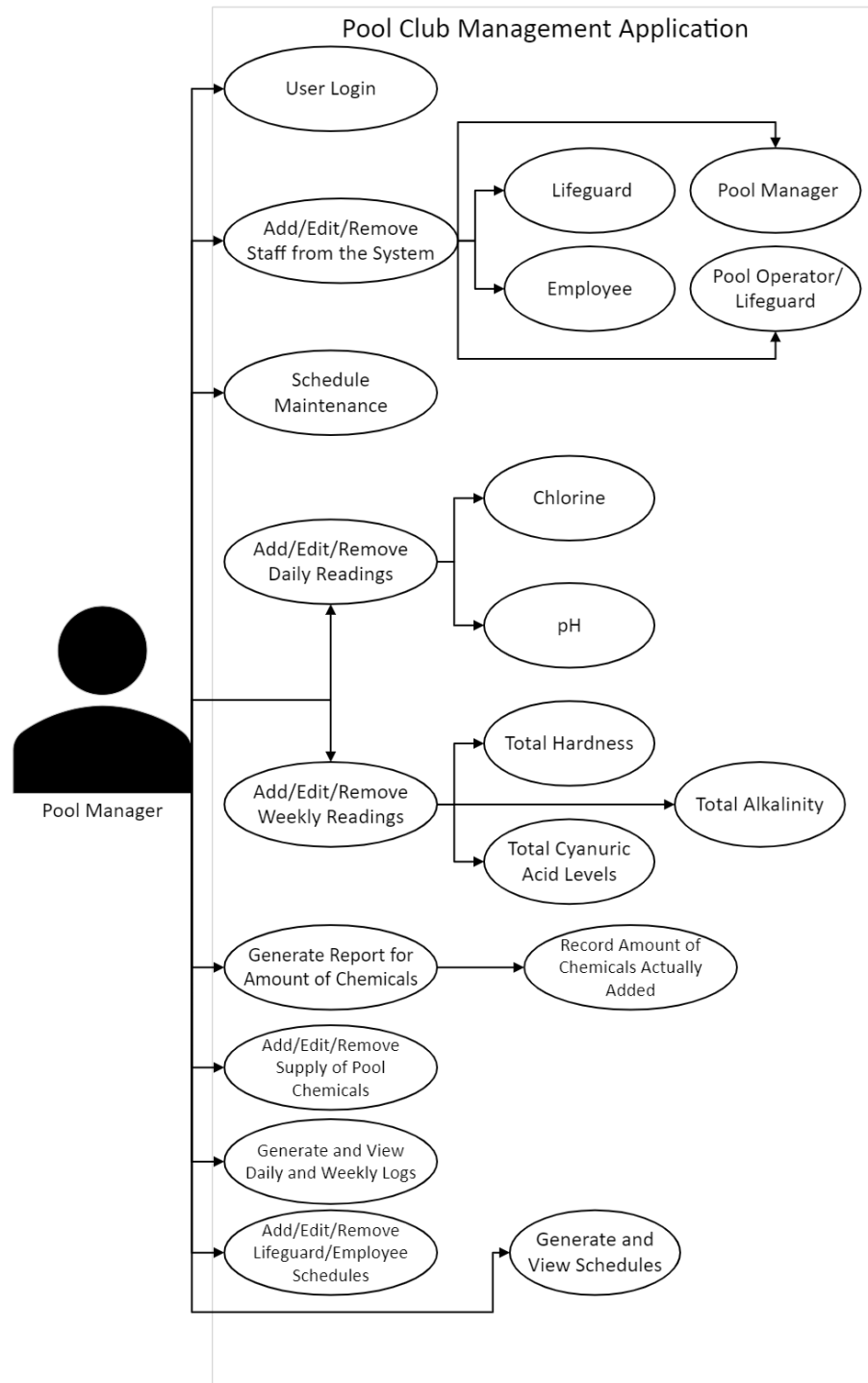
There are a number of learning opportunities tied to this project. First, I will learn how to integrate Python code with MySQL databases. This is done from the belief that learning about cross-language and cross-platform communication is useful so that certain functions already completed in one language can be reused in the context of another language or platform. This saves the software engineer time that would have otherwise been wasted on recreating certain features from scratch. It also helps one better understand how different types of software can communicate with one another in order to avoid compatibility issues. I will also learn how to craft a GUI interface using Python tkinter, a part of Python I have not yet interacted with, but I feel would be useful given the demand for Python that I have seen in various job announcements online. In addition, I would like to work on a project involving the Python programming language due to said previous demand for it.

As for the scope of the project and the size, I believe it is appropriate for a graduate case study project due to the involvement of system integration between different software tools and technologies that allow the application to fully unleash its true potential. It is also a worthy project since pool maintenance has a lot of moving parts and different variables attached to it depending on whether or not you have a public or private pool, much like how different maintenance operations have different needs and data needing to be managed depending on the type of maintenance being tracked.

Preliminary Design Plan

Use Cases and Use Case Diagrams

1. User login
2. Add, edit, or remove staff from the system.
 - a. Pool Manager
 - b. Pool Operator/Lifeguard
 - c. Lifeguard
 - d. Employee
3. Schedule maintenance
4. Add, edit, or remove daily readings.
 - a. Chlorine
 - b. pH
5. Add, edit, or remove weekly readings.
 - a. Total hardness
 - b. Total alkalinity
 - c. Total cyanuric acid levels
6. Generate report of chemicals required to meet regulatory levels.
 - a. Record amount of chemicals actually added to the pool.
7. Add, edit, or remove supply of pool chemicals.
8. Generate and view daily and weekly logs.
9. Add, edit, or remove lifeguard/employee schedules.
10. Generate and view schedules.

*Figure 25 Pool Manager Use Case Diagram*

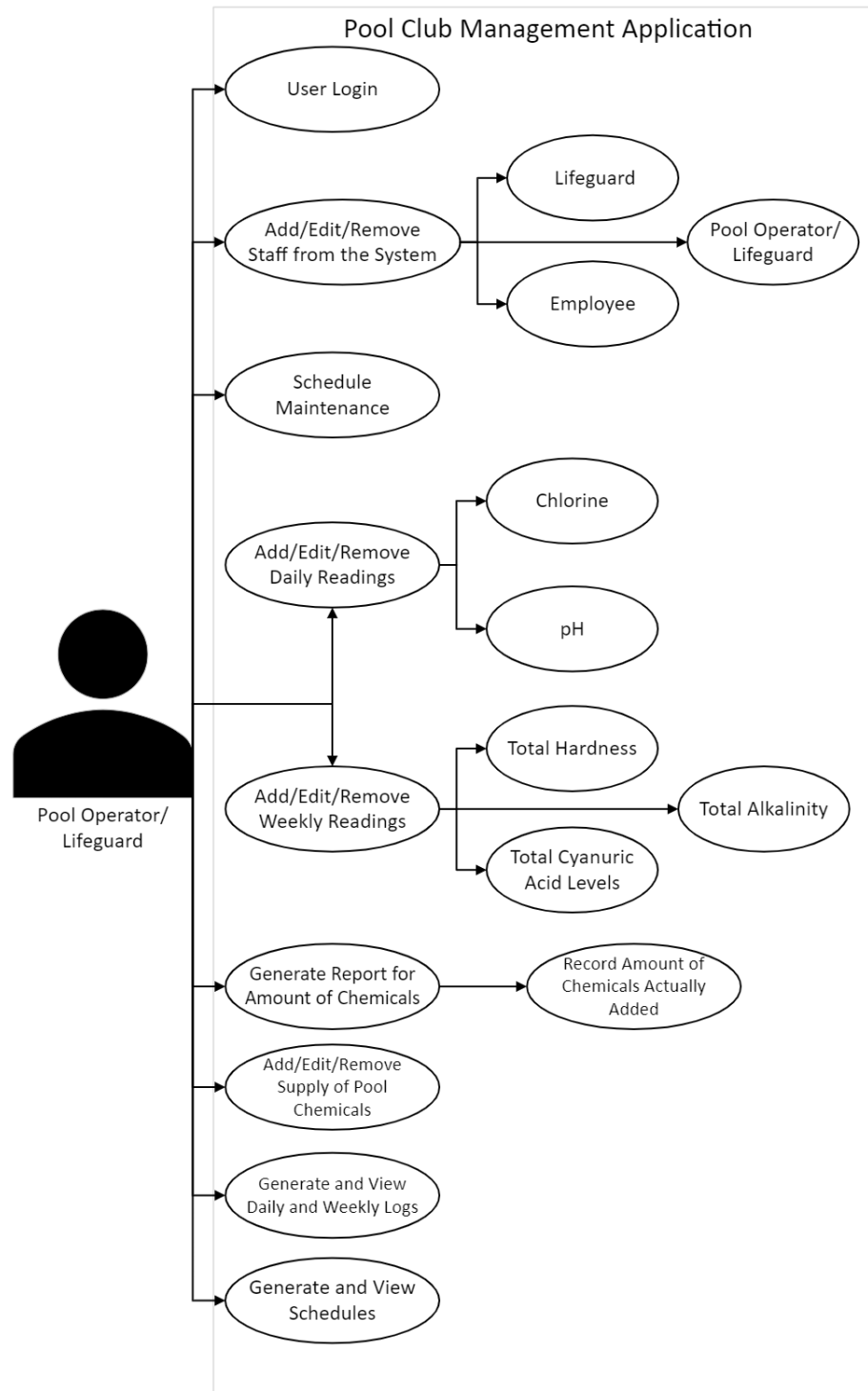


Figure 26 Pool Operator/Lifeguard Use Case Diagram

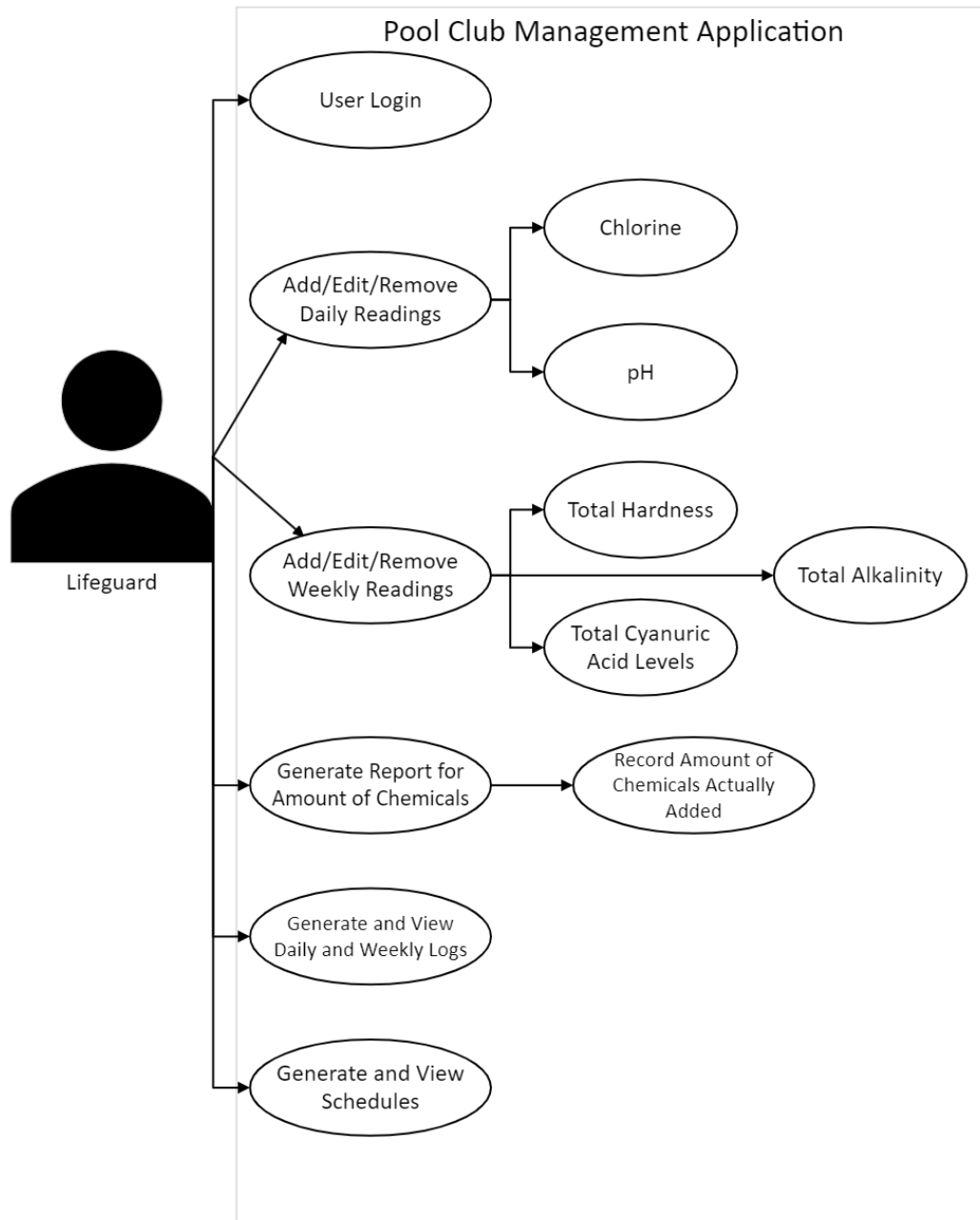


Figure 27 Lifeguard Use Case Diagram

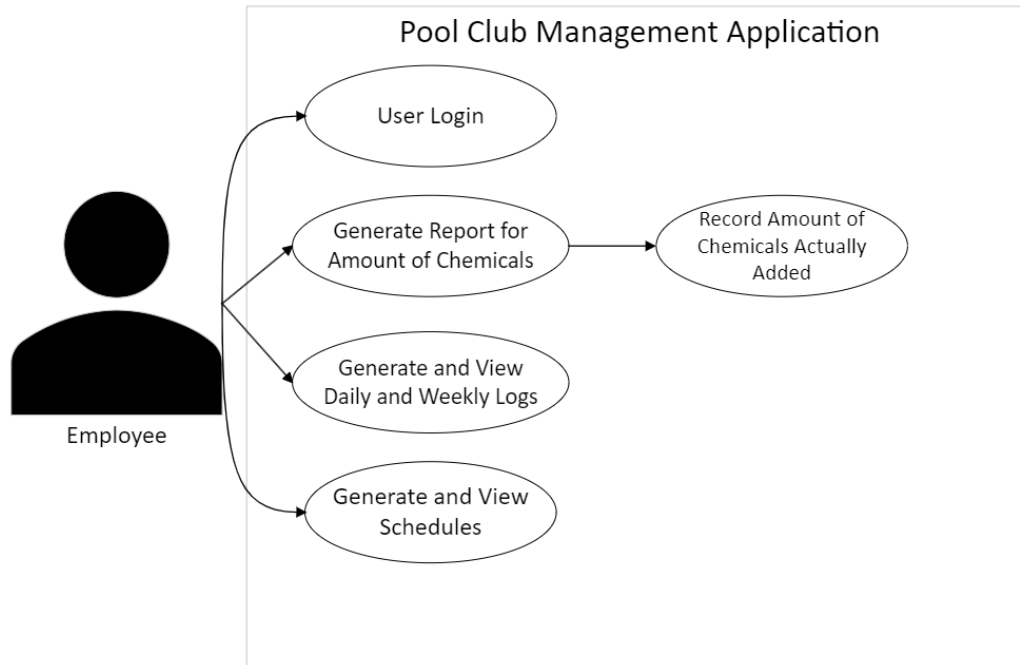
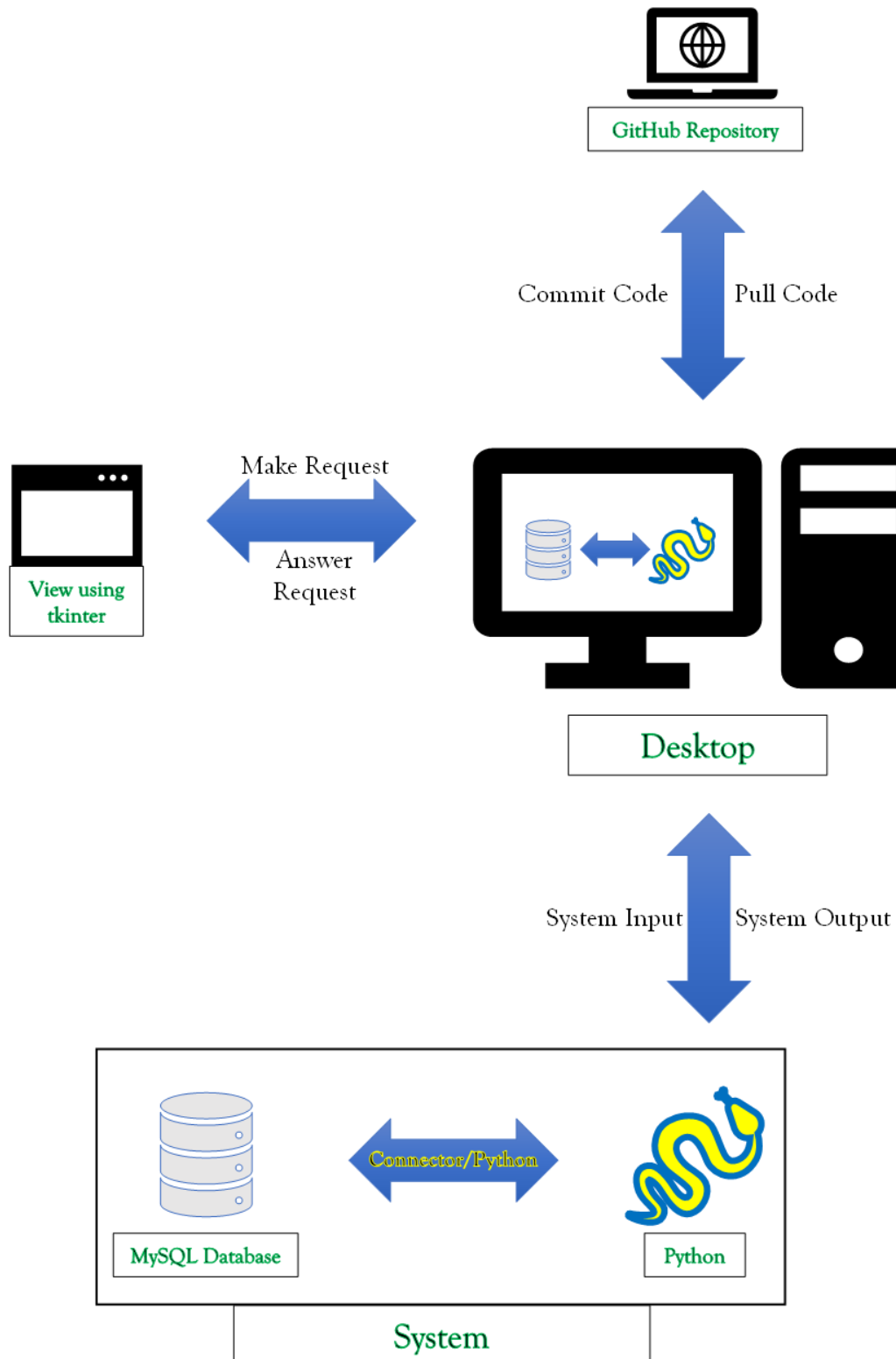


Figure 28 Employee Use Case Diagram

Preliminary Technology Framework/Platform

Technology	Purpose	Description
Python	The main coding language that the program will be written in.	A common and reliable programming language.
Python tkinter	Used in the creation of a Python GUI interface.	Python's GUI interface editor.
Python unittest	Used to test the Python code for different inputs.	A testing framework/platform for Python, much like JUnit is for Java.
MySQL	Used to build the database and where it is stored.	A database management software.
MySQL Connector/Python	Used as a bridge between the MySQL database and the Python GUI interface.	A connector tool allowing MySQL and Python to communicate.
GitHub	Used for configuration management (storing different versions of the same code in a repository).	An online repository for sharing and editing different versions of the same code base.

Preliminary Architectural Plan



Preliminary Schedule

Provide a preliminary schedule. The schedule needs to be at least 140 hours. The schedule needs to be granular and should contain activities related to regular meetings with faculty advisor.

Activities	Estimated Time (in hours)	Dates
Requirements and Research	10	08/28/2023
Setup Development Environment	10	09/04/2023
Advisor Meeting #1	1	09/11/2023
Defining Database Schema	10	09/11/2023
Create the Entity Relationship (ER) Diagram	8	09/18/2023
Advisor Meeting #2	1	09/25/2023
Analyze & Define Processes	10	09/25/2023
User Interface Diagram	8	10/02/2023
Advisor Meeting #3	1	10/09/2023
Design Code Modules	10	10/09/2023
Coding	10	10/16/2023
Advisor Meeting #4	1	10/23/2023
Coding	10	10/23/2023
Coding	10	10/30/2023
Advisor Meeting #5	1	11/06/2023
GUI Interface Mockup	8	11/06/2023
UI Code	10	11/13/2023
Unit and System Testing	10	11/20/2023
Advisor Meeting #6	1	11/27/2023
User Manual	8	11/27/2023
Case Study Draft Report	10	12/04/2023
Case Study Final Report due and Deliverables turned in		12/12/2023
Final Presentation and Presentation Recording	1	12/15/2023
Total	149	

Reading list

- [1] GitHub, “Let’s build from here,” GitHub, [Online], <https://github.com/> (accessed Jun. 27, 2023).
- [2] L. Barbosa and A. Hora, “How and Why Developers Migrate Python Tests,” in *2022 IEEE International Conference on Software Analysis, evolution and reengineering (saner)*, 2022, pp. 538–548
- [3] MySQL.com, “MySQL Connector/Python Developer Guide,” MySQL, [Online], <https://dev.mysql.com/doc/connector-python/en/> (accessed Jun. 28, 2023).
- [4] Python Software Foundation, “Python 3.11.4 documentation,” 3.11.4 Documentation, [Online], <https://docs.python.org/3/> (accessed Jun. 28, 2023).
- [5] Python Software Foundation, “tkinter - Python interface to TCL/TK,” Python documentation, [Online], <https://docs.python.org/3/library/tkinter.html> (accessed Jun. 28, 2023).
- [6] Python Software Foundation, “unittest — Unit testing framework,” Python documentation, [Online], <https://docs.python.org/3/library/unittest.html> (accessed Jun. 27, 2023).
- [7] M. Reichardt, M. Gundall, and H. D. Schotten, “Benchmarking the Operation Times of NoSQL and MySQL Databases for Python Clients,” in *IECON 2021 - 47th annual conference of the IEEE industrial electronics society*, 2021
- [8] B. A. Meier, *Python Gui Programming Cookbook: Develop Beautiful and Powerful GUIs Using the Python Programming Language*, Second. Birmingham, UK: Packt Publishing, 2017.

I, _____ *Evan Lyle* _____ propose to complete this project during the __*fall*__ semester of __*2023*__ and understand that this project and its derived materials (e.g., source code, written reports, presentation slides) are to reflect my own work, unless explicitly and appropriately referenced. Furthermore, I understand that plagiarism or other unattributed use of material not written by me is completely unacceptable, and **will be considered sufficient cause for a failing grade on the project.** For additional information on academic integrity policy at Towson University, I will visit www.towson.edu/provost/resources/studentacademic.asp.

Student's Signature: *Evan Lyle*

Instructor's Signature: _____

Graduate Program Director's Signature: _____

Appendix B – Requirements Document

The Requirements Document can be found on GitHub using the following link:

https://github.com/elyle1/pool_club_management/blob/main/Requirements%20Document.docx

Appendix C – API Documentation

The API Documentation can be found on GitHub using the following link:

https://github.com/elyle1/pool_club_management/blob/main/Pool%20Club%20Management%20Software%20E2%80%93%20API%20Documentation.docx

Appendix D – User Manual

The User Manual can be found on GitHub using the following link:

https://github.com/elyle1/pool_club_management/blob/main/Pool%20Club%20Management%20Software%20User%20Manual.docx

Appendix E – Source Code

The Source Code can be found on GitHub using the following link:

https://github.com/elyle1/pool_club_management